CS 5/7320
Artificial Intelligence

Making Simple Decisions:
Decision Networks

AIMA Chapter 16

Introduction slides by Michael Hahsler

Decision network slides by
Dan Klein and Pieter Abbeel

Online Material

Decision-theoretic Agents

Recap

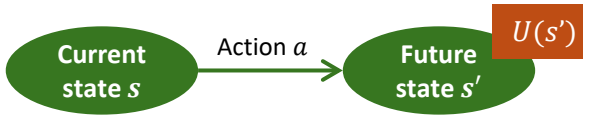
- **Agents based on logic**: Cannot deal with uncertainty, conflicting goals, etc.
- **Goal-based agents**: Can only assign goal/not goal to states and find goal states.

- A decision-theoretic agent is a special type of a **utility-based agent** based on decision theory.

  Approach:
  - Assign a **utility** value to each state.
  - Utility is related to the external performance measure (in the PEAS description).
  - Uses **probabilities** to assess its current state (for partial observability) and the results of actions (for stochastic environments).
  - A rational agent optimizes the **expected utility**.

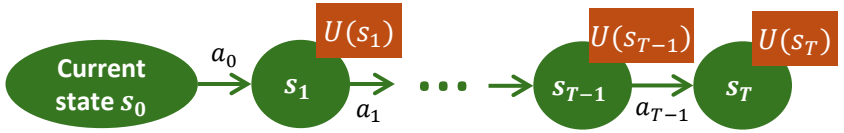
Decision theory =
Probability theory (evidence & belief)
+
Utility theory (want)

Simple vs. Complex Decisions



- We make the same decision frequently + making it once does not affect future decisions. This means we have an **episodic environment**.
- May have a **stochastic** environment (e.g., with non-deterministic actions or probabilistic transitions).
- May be **partially observable**.

We focus on making simple decisions for now!



- **Sequential decision making**: The agent's utility depends on a sequence of decisions.
- Search, planning, and playing games we have covered so far are examples of such problems.
- To solve this with decision theory requires different methods: **Markov Decision Processes (MDP)**

Utility

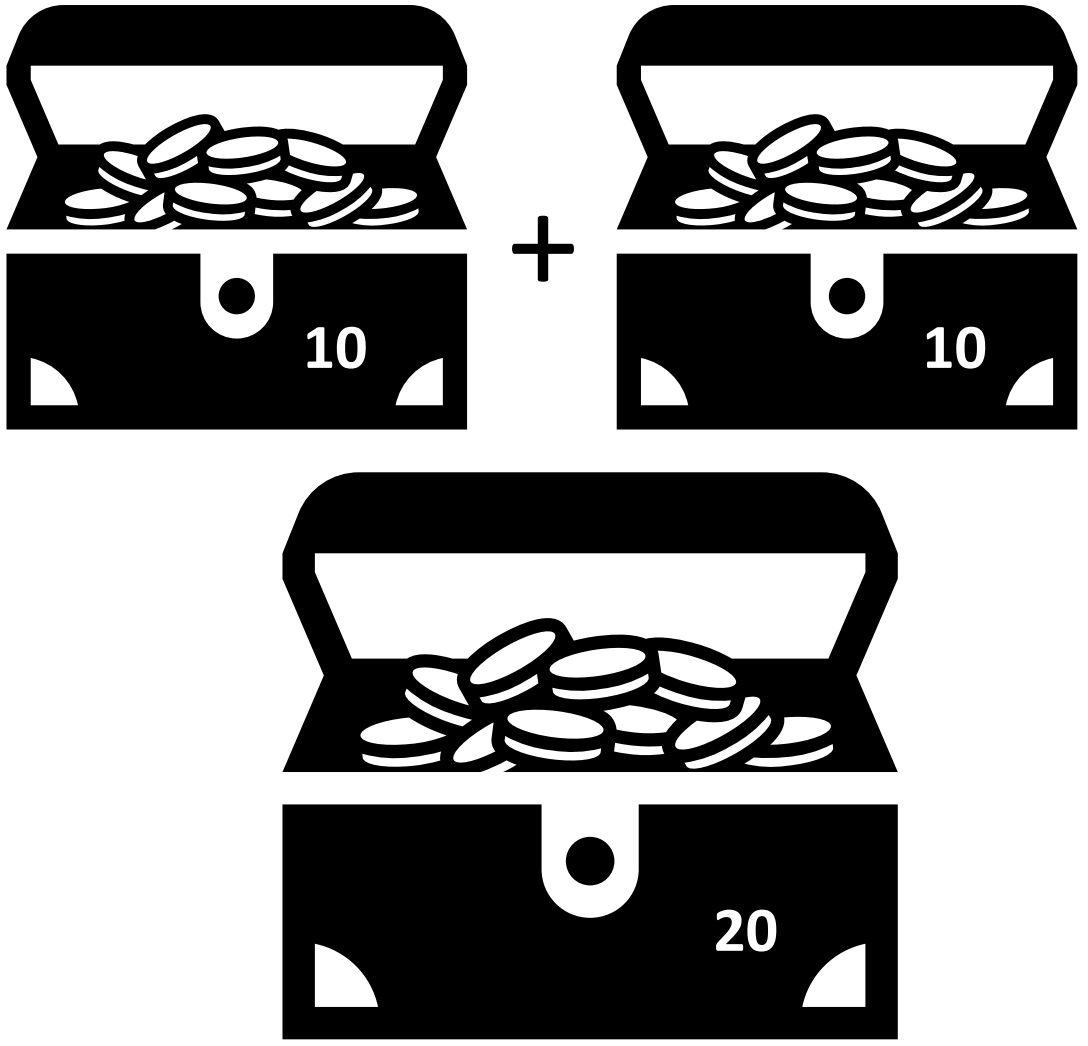
- A utility function $U(s)$ expresses the desirability of being in state s.
- Utility functions are derived from preferences:

$$U(A) > U(B) \Leftrightarrow A \succ B \quad \text{... Preference}$$

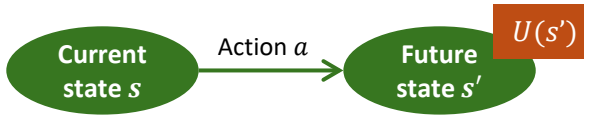
and

$$U(A) = U(B) \Leftrightarrow A \sim B \quad \text{... Indifference}$$

- It is often enough to know an **ordinal utility function** representing a **ranking** of states to make decisions like moving to the better state.
- To use expectation, we need a **cardinal utility function** where the number represents levels of absolute satisfaction.
  That is:
  - $\mathrm{U}(0) \sim$ nothing
  - $2 \times U$ is twice as good as U



Expected Utility of an Action Under Uncertainty



We need:

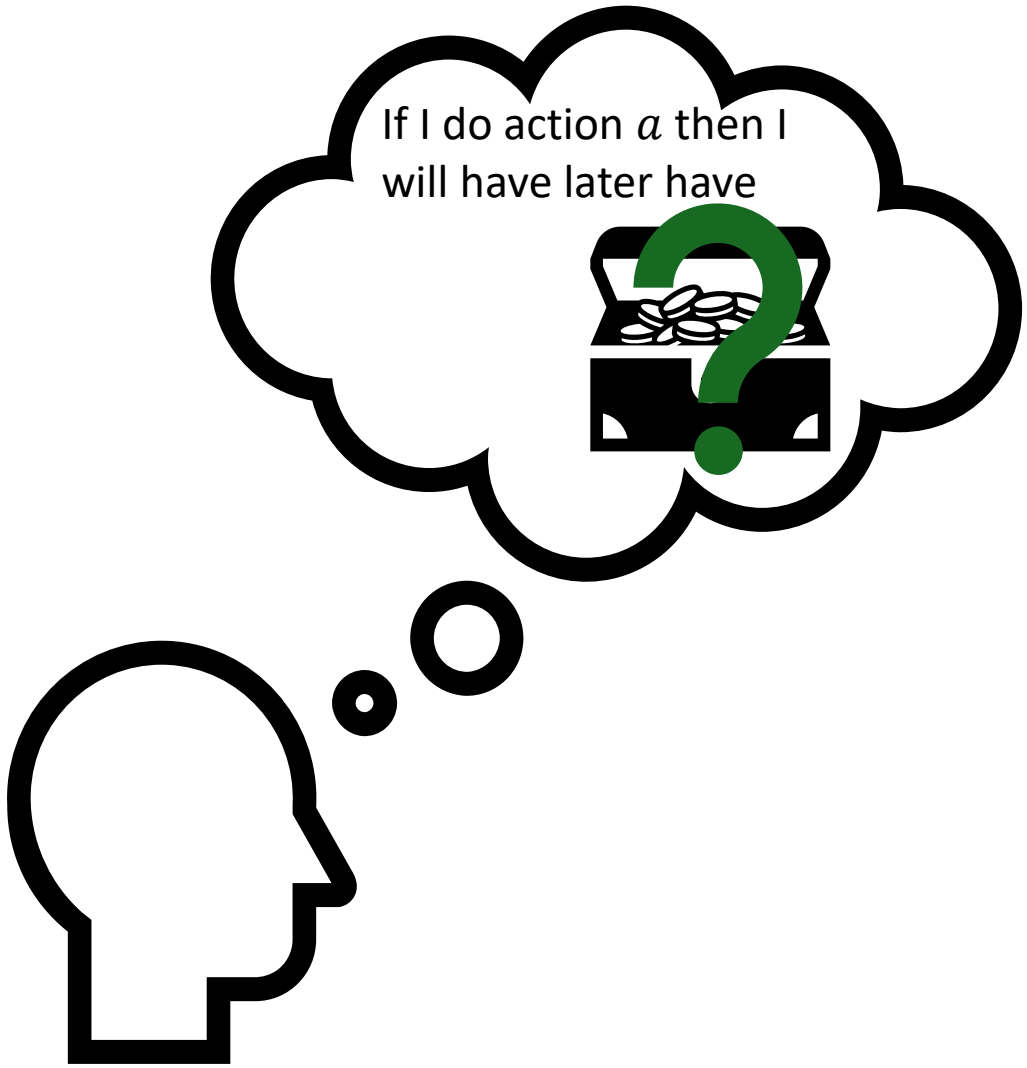
- The probability distribution $P(s)$, that the current state is s.
- The transition model specified as transition probabilities $P(s'|s,a)$ of getting from s to s' given action a.
- A **cardinal utility** function $U(s')$.

The probability that action a will get us from s to a future state s' is

$$P(s') = \sum_{s} P(s)\, P(s'|s,a)$$

The expected utility of action a over all possible future states is

$$EU(a) = \sum_{s'} P(s')\, U(s') = \sum_{s'} \sum_{s} P(s)\, P(s'|s,a)\, U(s')$$



Principle of Maximum Expected Utility (MEU)

Given the expected utility of an action

$$EU(a) = \sum_{s'} \sum_{s} P(s)\, P(s'|s,a)\, U(s')$$

A decision-theoretic agent chooses the action that maximizes the expected utility:

$$a^* = \text{argmax}_a\, EU(a)$$

Issue:

- $P(s'|s,a)$ is a very large table if we have many states.

Possible solution:

- Use a Bayesian network with a factored state representation considering conditional independence between random variables describing the state. This leads to **Decision Networks**.

Decision Networks

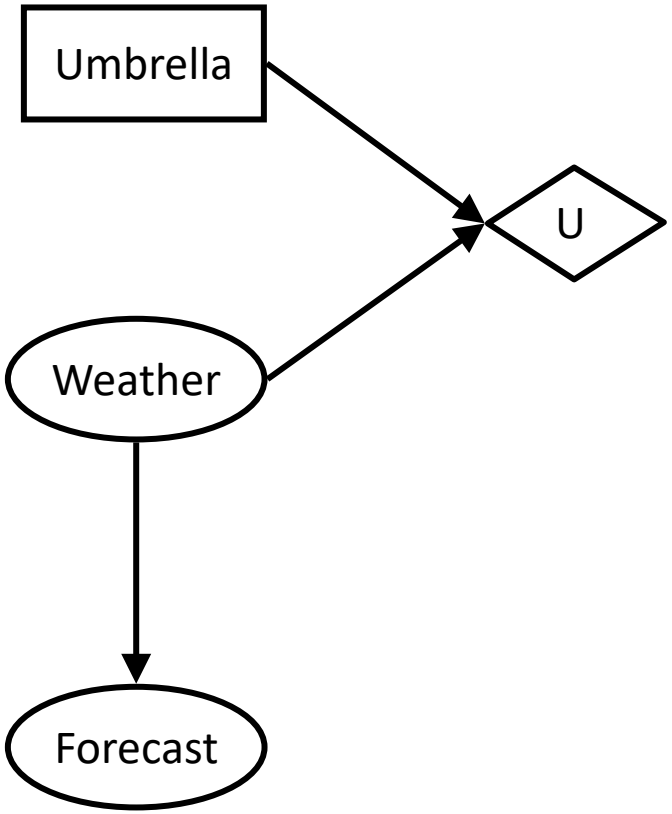
Using Bayesian Networks to calculate the Expected Utility of Actions

The slides are based on slides by Dan Klein, Pieter Abbeel, Sergey Levine, with some materials from A. Farhadi (http://ai.berkeley.edu)

Definition: Decision Networks

Decision networks

- Bayesian network with additional nodes for utility and actions.
- Allows to specify the joint probability in a compact way using conditional independence.
- Calculate the expected utility for each possible action and select the best one.



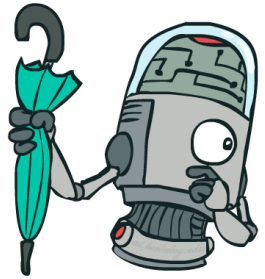
Node types

Chance nodes (oval): Random variables in BNs

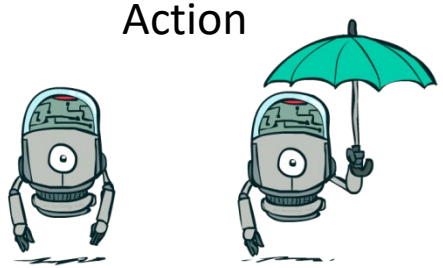
Action nodes (rectangle): Cannot have parents, act as observed evidence

Utility node (diamond): Depends on action and chance nodes

Example: Decision Networks



Action



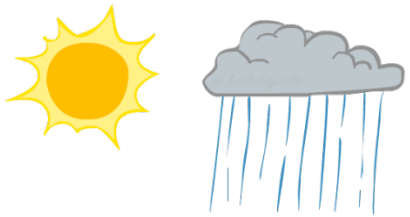
Umbrella

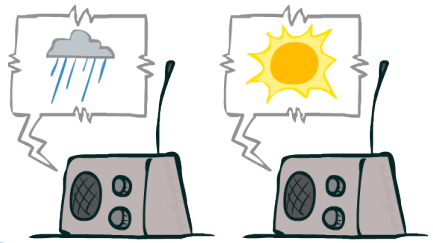
U

Weather

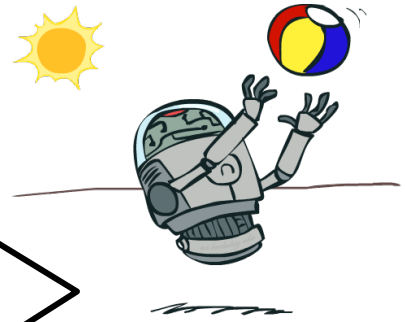
Forecast

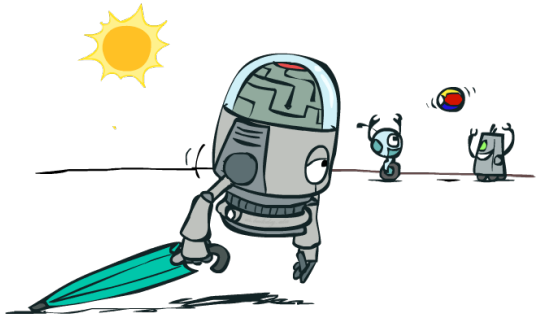
Random Events

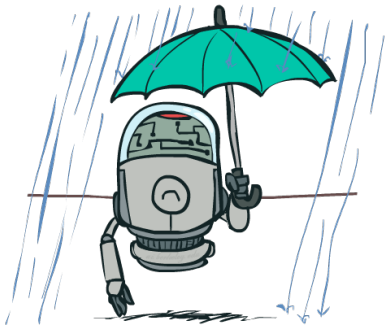




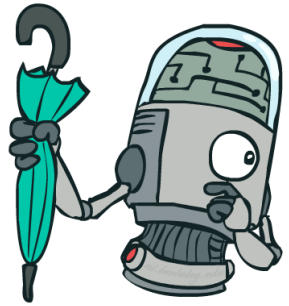
Utility

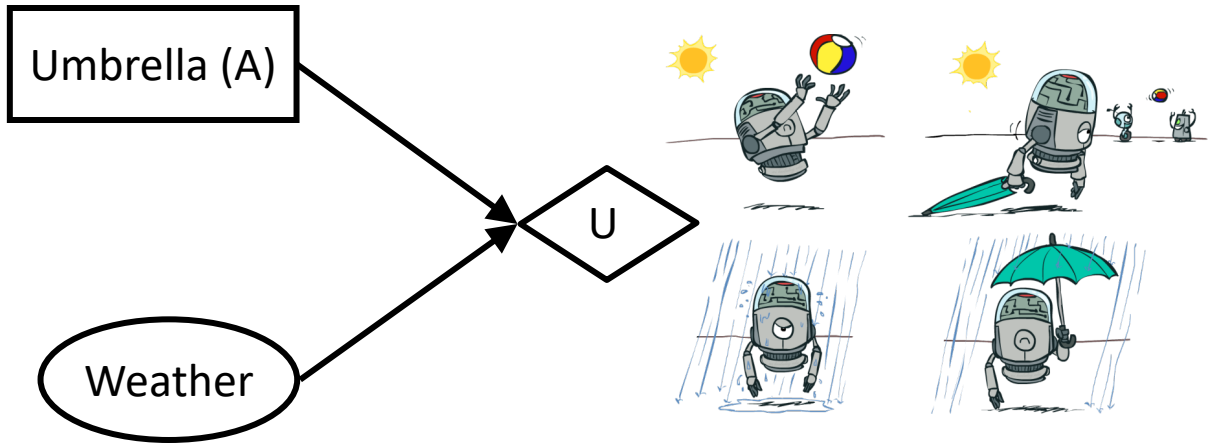






Decision Network without Forecast





W	$P(W)$
sun	0.7
rain	0.3

A	W	$U(A,W)$
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70

Action: Umbrella = leave

$$\mathrm{EU}(\mathrm{leave}) = \sum_w P(w)U(\mathrm{leave}, w)$$
$$= 0.7 \cdot 100 + 0.3 \cdot 0 = 70$$

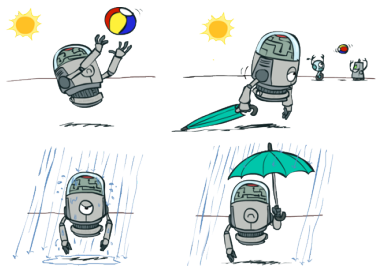
Action: Umbrella = take

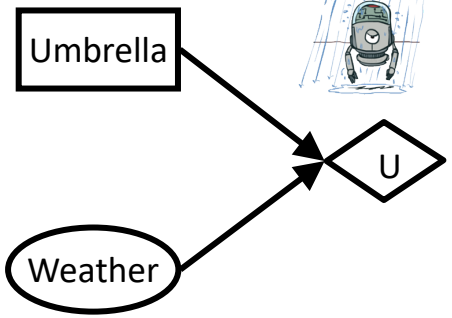
$$\mathrm{EU}(\mathrm{take}) = \sum_w P(w)U(\mathrm{take}, w)$$
$$= 0.7 \cdot 20 + 0.3 \cdot 70 = 35$$

Optimal decision a^* = leave

$$\mathrm{MEU}(\emptyset) = \max_a \mathrm{EU}(a) = 70$$

Decisions as Outcome Trees





{} … no evidence available

Action (agent picks one)

{}

take

leave

Weather | {}

Random Event (calculate expectation)

Weather | {}

sun

rain

sun

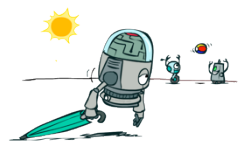
rain

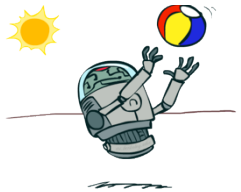
U(t,s)

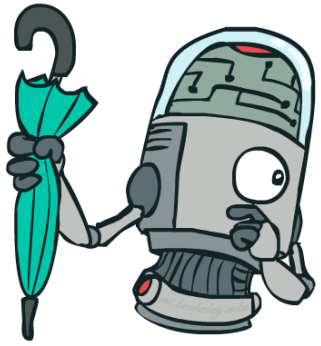
U(t,r)

U(l,s)

U(l,r)

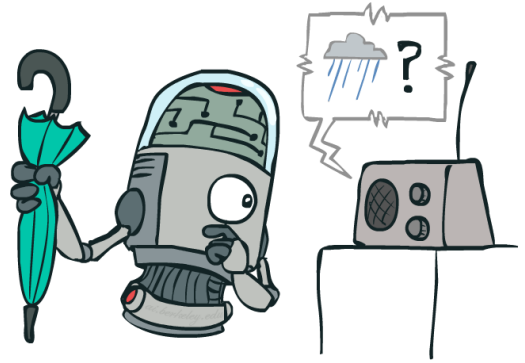


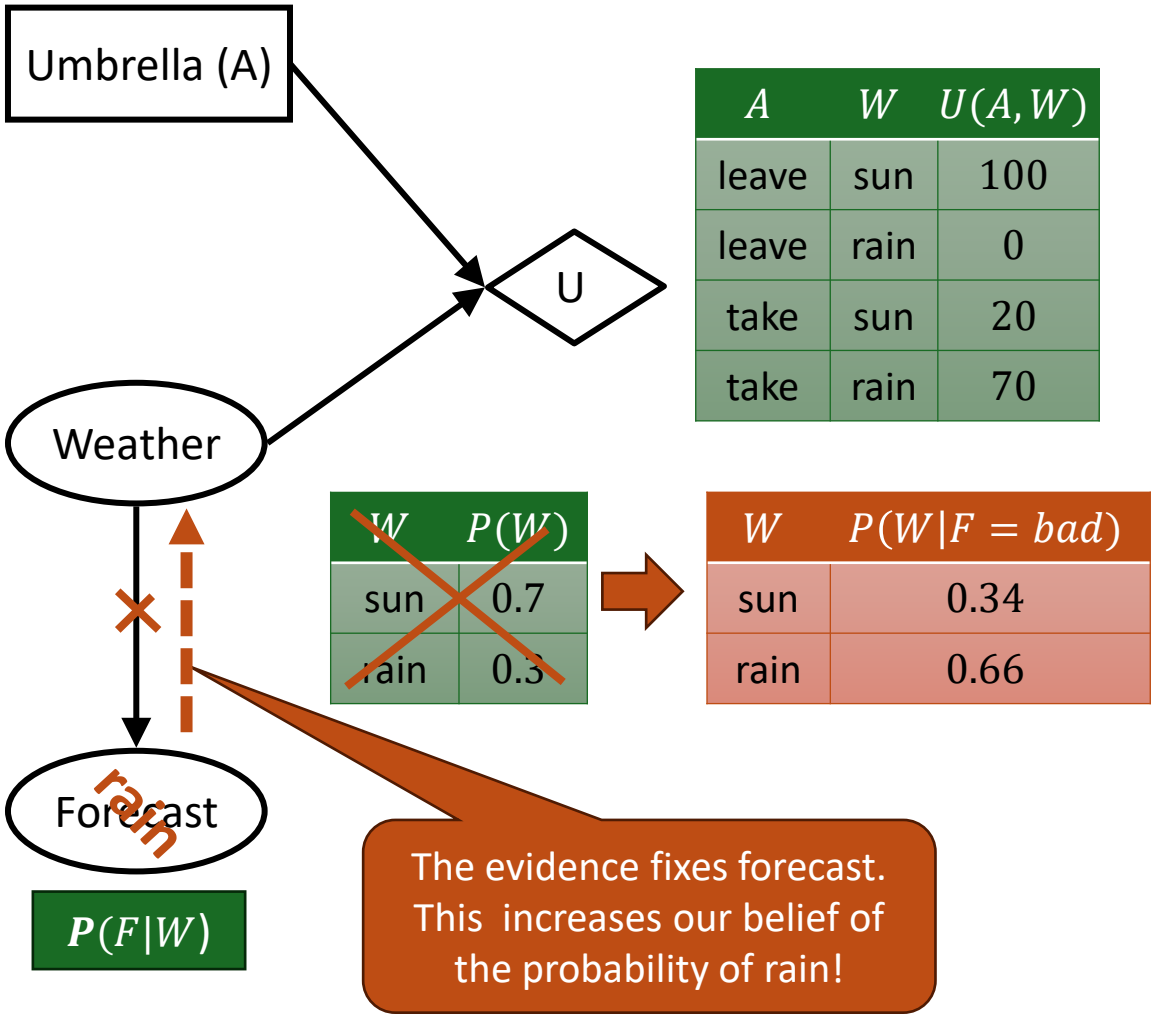




Traverse the tree and pick the actions that have the highest expected utility.

Decision Network with Bad Forecast





A	W	$U(A,W)$
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70

W	$P(W)$
sun	0.7
rain	0.3

W	$P(W\|F=bad)$
sun	0.34
rain	0.66

Action: Umbrella = leave

$$\mathrm{EU}(\text{leave}|\text{bad}) = \sum_w P(w|\text{bad})U(\text{leave}, w)$$
$$= 0.34 \cdot 100 + 0.66 \cdot 0 = 34$$

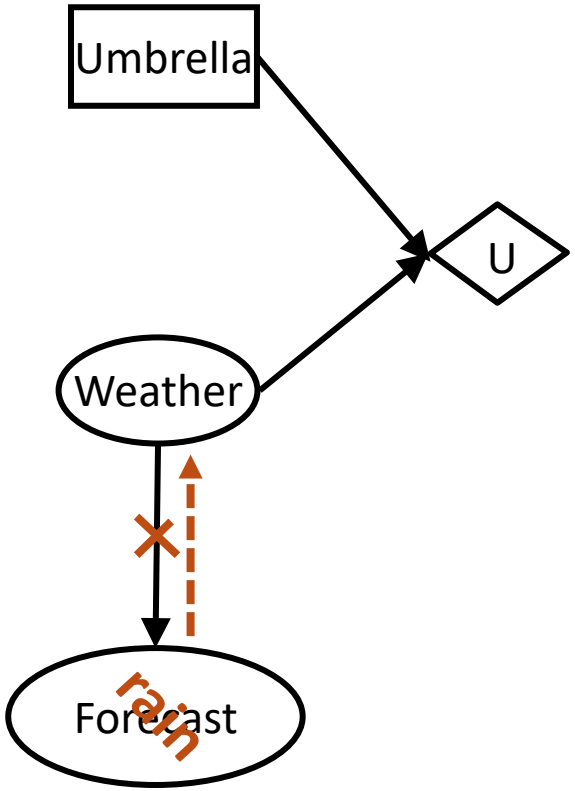
Action: Umbrella = take

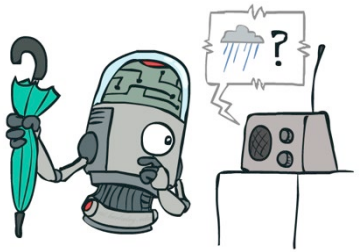
$$\mathrm{EU}(\text{take}|\text{bad}) = \sum_w P(w|\text{bad})U(\text{take}, w)$$
$$= 0.34 \cdot 20 + 0.66 \cdot 70 = 53$$

Optimal decision a^* = leave

$$\mathrm{MEU}(F = \text{bad}) = \max_a \mathrm{EU}(a|\text{bad}) = 53$$

Decisions as Outcome Trees with Evidence





{b} … evidence of bad weather increased the probability of rain

Action (agent picks one)

{b}

take

leave

W | {b}

Random Event (calculate expectation)

W | {b}

sun

rain

sun

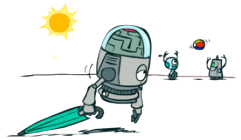
rain

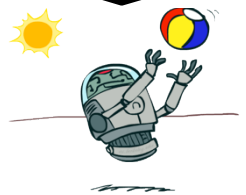
U(t,s)

U(t,r)

U(l,s)

U(l,r)





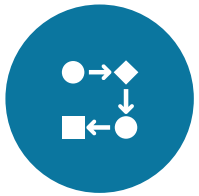
Traverse the tree and pick the actions that have the highest expected utility.

Conclusion

Decision networks are an extension of Bayesian networks that **add actions and utility** to compactly specify the joint probability.

The network is used to **calculate the expected utility of actions**.

Decision networks can be used to make simple repeated decisions in a **stochastic, partially observable, and episodic environment**.



Sequential decision-making deals with decisions that influence each other and are made over time. This is a more complex decision problem and needs different methods like **Markov Decision Processes.**